



Turbulent Flow Simulations of the NASA Common Research Model using the Immersed Boundary Method with a Wall Function

Yoshiharu Tamaki¹ and Taro Imamura²
The University of Tokyo, Tokyo, 113-8656, JAPAN

The transonic turbulent flows around the NASA Common Research Model is simulated to investigate the capability of the Cartesian-grid-based flow solver UTCart in 3D high Reynolds number flow simulations. Firstly, the framework of UTCart is explained. UTCart consists of an automatic Cartesian grid generator and a compressible flow solver. The immersed boundary method with a turbulent wall function is used in the flow solver to reproduce the turbulent boundary layer. The updates for UTCart related to the boundary heat flux and the calculation method for the cutoff distance are also described. Using UTCart, the medium grid (approx. 50 million cells) around the NASA-CRM is generated in 43 minutes. Compared with the grid converged value, the drag coefficient at the cruise condition has 24 drag count (8%) and 16 drag count (5%) differences in the medium and fine grid (approx. 97 million cells) results, respectively. Although the drag coefficient is slightly overestimated, all the component-wise aerodynamic coefficient shows a consistent trend of grid convergence. Furthermore, the qualitative flow features including flow separation at high angles of attack show a fair agreement with the experimental data and the computational results on the conventional body-fitted grids.

Nomenclature

α	=	angle of attack
c	=	local chord
C_D	=	total drag coefficient
$C_{D,p}$	=	pressure drag coefficient
$C_{D,v}$	=	viscous (friction) drag coefficient
C_L	=	lift coefficient
C_m	=	pitching moment coefficient
c_p	=	specific heat at constant pressure
C_p	=	pressure coefficient
d	=	wall distance
p	=	pressure
T	=	temperature
u_τ	=	friction velocity
τ_{ij}	=	total stress tensor
ν	=	kinematic viscosity
ν_t	=	kinematic eddy-viscosity
$\tilde{\nu}$	=	working variable in the Spalart-Allmaras turbulence model
ρ	=	density

Subscripts

IP	=	image point quantity
FC	=	face center quantity

¹ Graduate Student, Department of Aeronautics and Astronautics, 7-3-1, Hongo, Bunkyo-ku, Student Member.

² Associate Professor, Department of Aeronautics and Astronautics, 7-3-1, Hongo, Bunkyo-ku, Senior Member.

I. Introduction

Hierarchical Cartesian grid methods¹⁻⁶ are attracting attention as the methods for rapid/automatic grid generation around 3D complex geometries. The methods are also suitable for aerodynamic design where many different configurations are tested. Furthermore, the grids contain only orthogonal cells, and thus they have advantages in the numerical accuracy and stability. However, the wall boundaries in the Cartesian grids are expressed as sets of step-wise facets. Accordingly, additional treatments for the wall boundaries are required to recover smooth surface geometries.

The immersed boundary methods^{5, 6} are widely used to treat the wall boundaries in the Cartesian grids. They can maintain the simplicity and robustness of the grid generation because the generated Cartesian grid can be used without any additional processing (e.g. cell-cutting or overlapping). The authors are also developing a Cartesian-grid-based flow solver UTCart (The University of Tokyo Cartesian-Grid-Based Automatic Flow Solver) that uses the immersed boundary method for the wall boundary condition. UTCart consists of an automatic Cartesian grid generator and a compressible flow solver, which enable quick and automated flow simulation in the aircraft design.

However, application of the immersed boundary methods to simulation of high Reynolds number flows is a challenging problem. This is because the Cartesian grids contain only isotropic cells, and thus very small cells are needed to resolve the viscous sublayer of the turbulent boundary layers. In 3D problems, the number of cells near the wall boundary significantly increases when the grid size decreases, and thus the requirement for the cell size becomes especially problematic.

To solve the problem described above, the authors have developed an immersed boundary method for the Reynolds-averaged Navier-Stokes (RANS) simulations⁶. This method calculates the skin friction using a turbulent wall function, and thus the near-wall cell size can be equivalent to the height of the log-layer of the turbulent boundary layer. It has been also reported in the study that the modification of the velocity and eddy-viscosity profiles near the wall are essential to maintain the balance of numerical fluxes. In the 2D simple validation studies (e.g., a subsonic flow around the NACA0012 airfoil), the skin friction produced by the proposed method shows good agreement with the reference results on body-fitted grids⁷, and a consistent trend of grid convergence is observed. The method is incorporated in UTCart to enable high Reynolds number flow simulations.

The purpose of this research is to investigate the capability of UTCart in 3D high Reynolds number flow simulations. As the benchmark problem, transonic flows around the NASA Common Research Model (NASA-CRM)⁸ are simulated. The computed results are compared with the experimental data⁹ and computational results by the unstructured flow solver FaSTAR³ on conventional body-fitted grids.

The structure of this paper is as follows. In Section II, the overview of the automatic grid generator of the UTCart is described. In section III, the flow solver of UTCart including the turbulence wall boundary condition is explained. A few updates from the method described in the reference⁶ are included, which are related to the boundary heat flux and the calculation method for the cutoff distance that is required in the turbulent boundary condition. The setting of the simulation of the NASA-CRM is described in Section IV. The results are shown and discussed in Section V. Finally, Section VI concludes this paper.

II. UTCart Grid Generator

The UTCart grid generator creates Cartesian grids based on self-similar tree structure, i.e., a quad-tree (2D) or oct-tree (3D). In UTCart, the cells are classified into three categories: wall, fluid and body cells. Cells that are intersecting with or including the input surface geometry are treated as the wall cells. The fluid cells are those completely outside of the object, and the other cells are treated as the body cells. In the flow calculation, only the fluid cells are used, and the boundary conditions are imposed at the center of the faces between the wall and fluid cells (point FC in Fig. 1) using the immersed boundary method. To determine the boundary condition, the image points (IP) are set in the fluid cells.

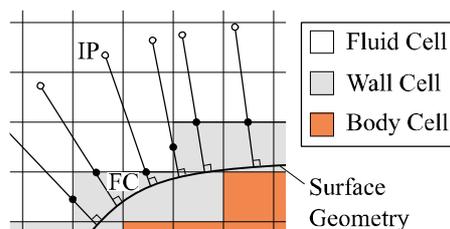


Fig. 1 Classification of cells around a surface geometry.

A. Initial Grid Generation

The surface geometry of the input objects is defined by sets of line segments (2D) or by STL files (3D). In the grid definition file, we set the cell size and the boundary condition for each input objects. As shown in Fig. 2, binary-tree structure with bounding boxes is created for each input object in order to find the nearest segment quickly. In addition, box/line sources for local grid refinement can be defined. For the box source, the corner coordinates of the box and the cell size are specified. For the line source, the start/end point, the radius, and the cell size are assigned.

In the grid generation, the entire computational domain is named root cell. If the input objects exist in the computational domain, the root cell is a wall cell (Fig. 3 (a)). Next, the wall cells and the cells inside the sources for local refinement are recursively divided into 4 (2D) or 8 (3D) cells until the cell size becomes smaller than the value specified for the object or the source (Fig. 3 (b)). The grid is then smoothed to limit the size ratio between the adjacent cells to 2 (Fig. 3 (c)). The number of the cells in the wall/smoothing layer can be specified separately, as shown in Fig. 4. Finally, the cells outside of the object are specified, and they are transferred into the flow solver (Fig. 3 (d)).

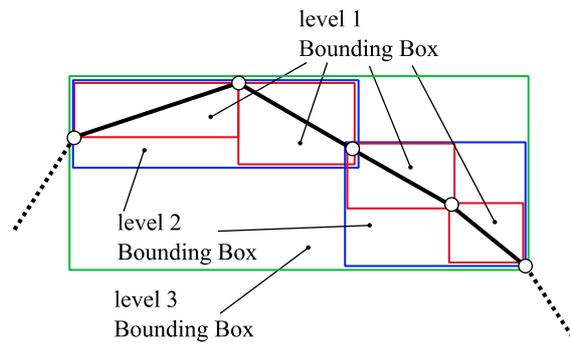


Fig. 2 Binary-tree structure for the input surface geometry.

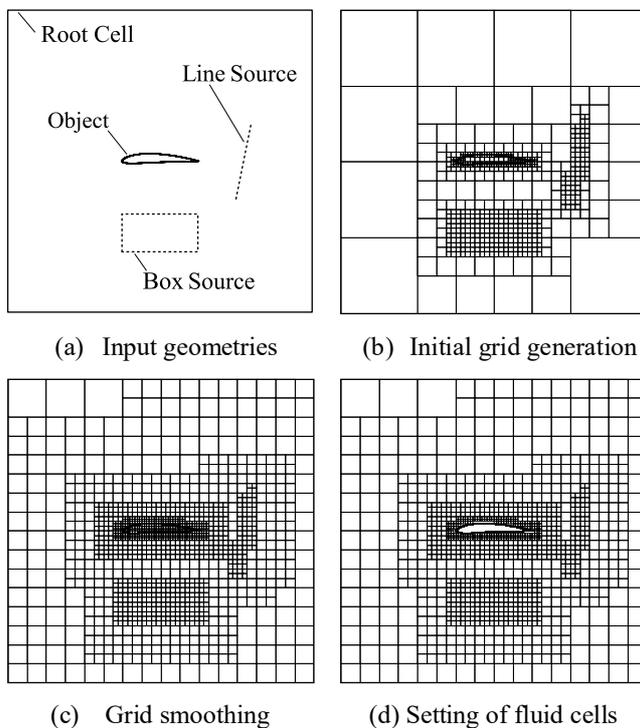


Fig. 3 Outline of the Cartesian grid generation.

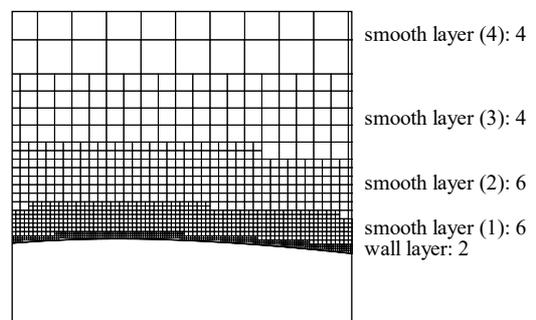


Fig. 4 Setting of wall/smooth layer.

B. Setting of Image Point

The IP is set on the wall normal line passing through point FC. The distance between the IP and wall is a constant value r_{IP} times the size of the cell that includes point FC. The minimum value of r_{IP} is $\sqrt{2}$ in 2D and $\sqrt{3}$ in 3D to assure that the IP exists in a fluid cell. To capture the boundary layer, a large value of r_{IP} should also be avoided. Therefore, r_{IP} is fixed 2 in this research. Furthermore, a non-slip condition is imposed when the IP is located inside the body. This situation occurs where the grid size is insufficient to resolve the surface features (e.g. a small gap).

III. UTCart Flow Solver

The flow solver of UTCart is based on the compressible Navier-stokes equation. The generated Cartesian grid is treated as an unstructured grid, and is discretized by the finite-volume method. The basic turbulent model is the Spalart-Allmaras (SA) model¹⁰ without the trip term (-noft2). The explicit Runge-Kutta method or implicit Lower-Upper Symmetric Gauss-Seidel (LU-SGS)¹¹/ Matrix-Free Gauss-Seidel (MFGS)¹² are available for the time integration. For the inviscid term, Roe's flux difference splitting¹³ or the SLAU scheme¹⁴ can be used with the second-order MUSCL¹⁵ or the fourth-order upwind-biased scheme¹⁶. The second-order MUSCL is also used in the convection term of the SA equation. For the viscous term, the second-order central difference with a correction term¹⁷ is used to avoid the even-odd decoupling. To calculate the aerodynamic force working on the immersed body, the integration method based on the numerical flux (see Appendix A) is used.

A. The Immersed Boundary Method

To determine the physical quantities at FC (see Fig. 1), 1D profiles between the IP and wall are assumed. Note that the quantities at IP are linearly interpolated from the surrounding cells using the method described in the reference⁵. In the explanation below, the values at the IP and FC are represented by the subscripts *IP* and *FC*, respectively. In addition, the y -coordinate is fixed to the wall normal direction for simplicity.

In RANS simulations, the boundary condition using a turbulent wall function⁶ is imposed at the wall boundary. In this method, the friction velocity u_τ is determined by the Newton iterations at first, so that the wall tangential velocity at the IP satisfies the wall function profile:

$$u_{t,IP} = u_\tau f_{wall}(y_{IP}^+), \quad (1)$$

where $y^+ = u_\tau d/\nu$, $f_{wall}(y^+)$ is the turbulent wall function, ν is the kinematic viscosity, and d is the distance from the wall. For the turbulent wall function $f_{wall}(y^+)$, the SA wall model¹⁸ is used to keep the consistency with the SA turbulence model. Next, the modified tangential velocity profile is assumed using a partial slip condition (see Fig 5 (a)):

$$u_t(y^+) = u_{t,IP} - \left\{ \frac{\partial f_{wall}}{\partial y^+}(y_{IP}^+) \right\} (y_{IP}^+ - y^+) u_\tau. \quad (2)$$

Note that the first derivative of the wall function can be calculated analytically⁶. Although the actual velocity profile in the turbulent boundary layer is a non-linear function, the modified profile is linear with respect to the wall distance. Thus, the modified velocity profile can be resolved on a Cartesian grid without layer cells. The boundary condition for the tangential velocity at point FC is calculated by substituting $y^+ = y_{FC}^+$ in eq. (2).

Furthermore, the eddy-viscosity profile must be modified to maintain the balance of the shear stress. From the boundary-layer approximation, the shear stress is constant with respect to the wall distance. In the modified velocity profile (2), the velocity gradient is also constant. Thus, the required condition for the eddy-viscosity is $\nu_t = \text{const.}$ between the IP and wall. In the SA turbulence model, the relationship between the working variable $\tilde{\nu}$ and the eddy viscosity μ_t is

$$\mu_t = \rho \tilde{\nu} f_{v1}, \quad (3)$$

where f_{v1} is the wall damping function. The SA turbulence model has a linear solution $\tilde{\nu} = \kappa u_\tau y$ ($\kappa = 0.41$) in the inner layer of a developed turbulent boundary layer. Thus, the wall-damping function f_{v1} in the SA turbulence model is modified as follows to realize the constant eddy-viscosity:

$$f_{v1} = \begin{cases} f_{v1,w} & (d < d_{cutoff}) \\ f_{v1,f} & (d \geq d_{cutoff}) \end{cases}, \quad (4)$$

$$f_{v1,w} = r_d \frac{(\chi r_d)^3}{c_{v1}^3 + (\chi r_d)^3},$$

$$f_{v1,f} = \frac{\chi^3}{\chi^3 + c_{v1}^3},$$

where $c_{v1}=7.1$, $\chi = \tilde{\nu}/\nu$ and $r_d = d_{cutoff}/d$. Here, the cutoff distance d_{cutoff} is equal to the distance between the IP and wall. The near-wall damping function $f_{v1,w}$ is proportional to the inverse of the distance d , and thus the eddy-viscosity becomes constant below the cutoff distance (Fig 5 (b)).

One modification is made on the temperature boundary condition. In the previous study, the near-wall temperature profile is given by the Crocco-Busemann relationship¹⁹ as

$$T_{FC} = T_{IP} + \frac{Pr^{\frac{1}{3}}}{2c_p} (u_{t,IP}^2 - u_{t,FC}^2), \quad (5)$$

where T is the temperature, the Prandtl number $Pr = 0.72$, and c_p is the specific heat at constant pressure. In the updated method, the temperature profile is also modified to match the modified velocity profile. Here, the normal temperature gradient is calculated by differentiating the Crocco-Busemann relationship as

$$\frac{dT}{dy} = -\frac{Pr^{\frac{1}{3}}}{c_p} u_t \frac{du_t}{dy}. \quad (6)$$

In the modified velocity profile (2), the normal gradient of the tangential velocity (du_t/dy) is constant between the IP and wall. The tangential velocity u_t is not constant, but $u_t \sim u_{t,IP}$ is assumed. This is because the velocity gradient in the log-layer of the turbulent boundary layer is generally small, and the modified velocity profile (2) gives only a small deviation from $u_{t,IP}$. Therefore, the temperature gradient (dT/dy) becomes nearly constant in the modified profile.

The boundary condition for the temperature at point FC should be consistent with the conditions described above. Thus, the temperature is

$$T_{FC} = T_{IP} - \left. \frac{dT}{dy} \right|_{IP} (y_{IP} - y_{FC}), \quad (7)$$

where the normal temperature gradient is calculated as

$$\begin{aligned} \left. \frac{dT}{dy} \right|_{IP} &= \frac{Pr^{\frac{1}{3}}}{c_p} u_{t,IP} \left. \frac{du_t}{dy} \right|_{IP} \\ &= \frac{Pr^{\frac{1}{3}}}{c_p} f_{wall}(y_{IP}^+) \left\{ \frac{\partial f_w}{\partial y^+}(y_{IP}^+) \right\} \frac{u_t^3}{\nu}. \end{aligned} \quad (8)$$

This boundary condition assumes a linear profile of the temperature gradient (Fig. 5 (c)). Here, the heat flux is

$$q_y = -c_p \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) \frac{dT}{dy}, \quad (9)$$

where Pr_t is the turbulent Prandtl number ($=0.90$). The eddy-viscosity is also modified to be constant between IP and wall, and thus, the heat flux becomes constant between the IP and wall.

The remaining boundary conditions are given as follows. The wall normal velocity at point FC is derived from a non-penetration condition at the wall:

$$u_{n,FC} = u_{n,IP} \frac{d_{FC}}{d_{IP}}. \quad (10)$$

The pressure is assumed to be constant in the boundary layer:

$$p_{FC} = p_{IP}. \quad (11)$$

The density is

$$\rho_{FC} = \frac{p_{FC}}{RT_{FC}}, \quad (12)$$

where R is the gas constant. In addition, the boundary condition for $\tilde{\nu}$ is derived from the near-wall solution of SA turbulence model as

$$\tilde{\nu}_{FC} = \kappa u_{\tau} d_{FC}. \quad (13)$$

The inviscid flux is computed by an upwind flux $\mathbf{F} = \mathbf{F}(\mathbf{q}_L, \mathbf{q}_{FC})$, where \mathbf{q}_L is the extrapolated quantities from the fluid cells, and \mathbf{q}_{FC} is the quantities at point FC given by eqs. (2, 10-13). The magnitude of the viscous stress τ at point FC is directly imposed as $\tau_{FC} = \rho u_{\tau}^2$. Furthermore, the heat flux is modified to be constant between the IP and wall. Thus, the boundary heat flux at point FC is calculated using the quantities at the IP as

$$q_{y,FC} = -c_p \left(\frac{\mu}{Pr} + \frac{\mu_{t,IP}}{Pr_t} \right) \left. \frac{dT}{dy} \right|_{IP}. \quad (14)$$

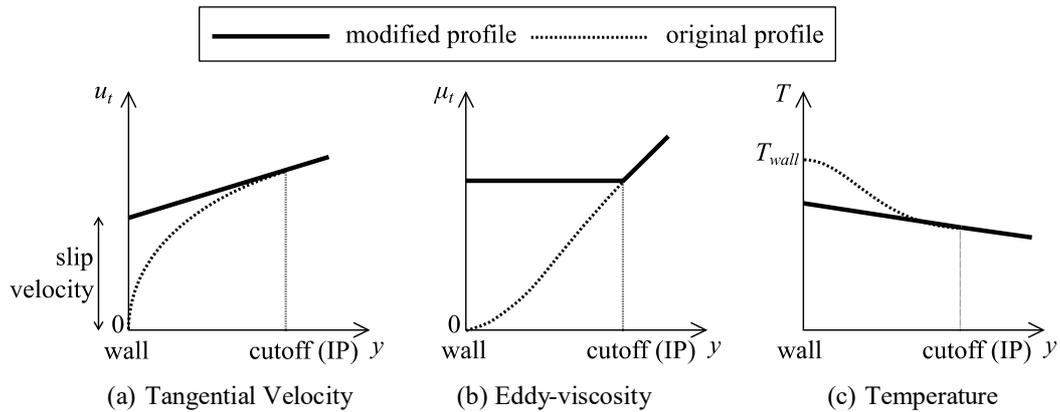


Fig. 5 Modified profiles of variables near wall.

B. Calculation of Wall/Cutoff Distance

In the SA turbulence model, the distance from the nearest wall is required in each cell. In UTCart, the levelset method²⁰ is used to calculate the distance. The distance is given by the solution of the equation for the distance function $\phi = \phi(x, y, z, \tau)$

$$\frac{\partial \phi}{\partial \tau} = 1 - \sqrt{\left(\frac{\partial \phi}{\partial x}\right)^2 + \left(\frac{\partial \phi}{\partial y}\right)^2 + \left(\frac{\partial \phi}{\partial z}\right)^2} \quad (15)$$

at the pseudo time $\tau \rightarrow \infty$. The boundary condition for eq. (15) is imposed at point FC by calculating the distance between point FC and wall directly.

In addition, the cutoff distance d_{cutoff} is required in eq. (4) to calculate the eddy-viscosity at each cells. The cutoff distance is equal to the IP distance (or r_{IP} times cell size) at the nearest wall. The cutoff distance is constant when the same cell size is assigned to all the objects in the computational domain. If there are multiple objects that have different cell sizes, however, all the cells in the domain need to know which object is the nearest. It is even more difficult to know the cell size at the nearest wall when a local refinement has a smaller cell size than that assigned to the objects.

Therefore, a new method to calculate the cutoff distance is developed using the advection equation that propagates the wall information to the domain. The cutoff distance must be constant in the wall normal direction. Here, the advection equation for the cutoff distance function $\psi = \psi(x, y, z, \tau)$ is

$$\frac{\partial \psi}{\partial \tau} + n_x \frac{\partial \psi}{\partial x} + n_y \frac{\partial \psi}{\partial y} + n_z \frac{\partial \psi}{\partial z} = 0, \quad (16)$$

where n_x, n_y, n_z is each component of the normal vector. The normal vector is calculated by differentiating of the levelset solution ϕ as

$$n_x = \frac{\partial \phi}{\partial x}, \quad n_y = \frac{\partial \phi}{\partial y}, \quad n_z = \frac{\partial \phi}{\partial z}. \quad (17)$$

The derivatives are calculated by the linear least-squares method. The wall boundary condition of the advection equation (16) is imposed at point FC as a Dirichlet boundary condition $\psi = r_{IP} \Delta x_L$, where Δx_L is the size of the cell including point FC. Also, a non-gradient boundary condition is imposed at the far-field. Equation (16) is spatially discretized by the finite-volume method as

$$\frac{\partial \psi_i}{\partial t} + \frac{1}{V_i} \sum_{j \in \epsilon_i} \tilde{F}_{ij} s_{ij} = 0, \quad (18)$$

where j denotes the index of the neighbour cells around cell i , V_i is the cell volume, \tilde{F}_{ij} is the numerical flux on the face between cells i and j , and s_{ij} is the face area. Note that summation is not taken in terms of the indexes i and j . The time is advanced by a the first-order Euler explicit method. To stabilize the computation, an upwind numerical flux is used. Here, the numerical flux at the face between cells i and j is

$$\tilde{F}_{ij} = \begin{cases} \psi_i \alpha_i & (\alpha_i > 0) \\ \frac{\psi_i + \psi_j}{2} \alpha_i & (\alpha_i \leq 0 \text{ and } \alpha_j > 0) \\ \psi_j \alpha_i & (\alpha_i \leq 0 \text{ and } \alpha_j \leq 0) \end{cases} \quad (19)$$

where the normal advection speed α is the inner product of the normal vector (n_x, n_y, n_z) and the face normal. Note that eq. (19) is the flux seeing from cell i , and the flux seeing from cell j can be different. This definition of the flux can stop the propagation when information from two directions intersects. Finally, the cutoff distance d_{cutoff} is given by the solution of the function ψ at $\tau \rightarrow \infty$.

For example, the computed cutoff distance around two airfoils in a 2D domain is illustrated in Fig. 6. The cell sizes assigned to the upper and lower airfoils are 0.04 and 0.02, respectively. In addition, a local refinement with $\Delta x = 0.01$ is located near the leading edge of the lower airfoil. It is confirmed that the computational domain is correctly divided into three domains by the cell sizes at the nearest wall, and that the calculated cutoff distance is r_{IP} times the cell size at the nearest wall.

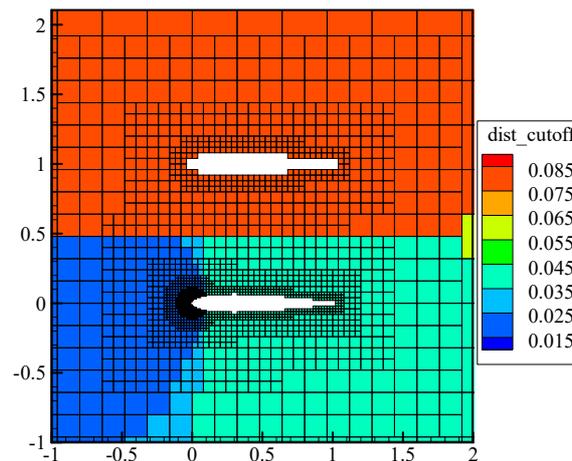


Fig. 6 Example of the computed cutoff distance around two airfoils including local refinement.

IV. Computational Settings

A. Problem Definition

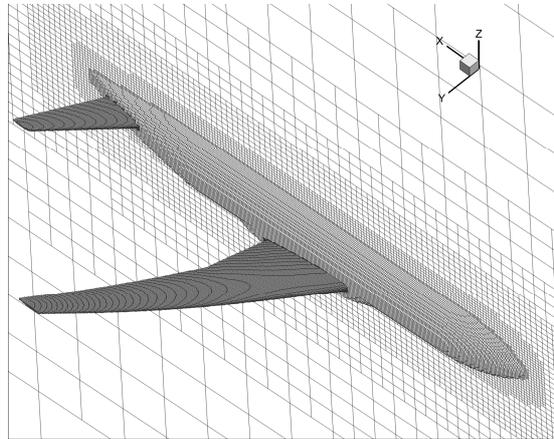
We simulate a half-cut model of the wing-body-tail configuration of the NASA-CRM⁸ with the tail incidence attack $i_H = 0$ deg. The geometry is initially designed for the benchmark problem in the NASA drag prediction workshop²¹. Recently, another workshop called the aerodynamic prediction challenge (APC)²² was held in Tokyo, 2015, and in Kanazawa, 2016, in order to improve the prediction accuracy of aerodynamic performance of an aircraft. Experimental study⁹ was also conducted in JAXA's transonic wind tunnel using a 2.16% model (the mean aerodynamic chord $c_{ref} = 151.31$ mm). The conditions of the calculation are adjusted to the experimental setting: the Reynolds number based on c_{ref} is 2.26×10^6 , the freestream Mach number is 0.847, the freestream temperature T_{ref} is 284 K, and the angles of attack α are -1.79, -0.62, 0.32, 1.31, 2.94, 3.55, 4.65, 5.72 deg. The aerodynamic deformation of the main wing is considered at each angle of attack according to the experimental data.

B. Computational Grid and Methods

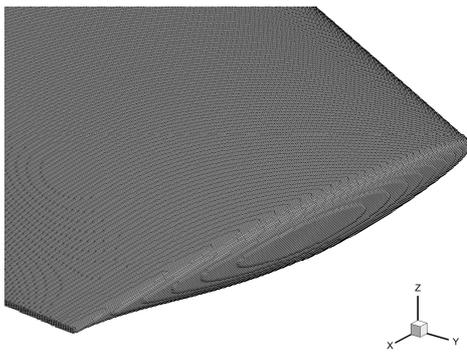
UTCART uses octree-based Cartesian grids (see Fig. 7). To investigate the trend of grid convergence, three grids with different resolution of cells are prepared, as described in Table 1. The length scale in the table is based on the actual scale of the NASA-CRM ($c_{ref} = 275.8$ in). The number of the smooth layer is changed as well as the grid size to obtain a fair grid convergence. Note that the total cell number slightly varies depending on the angle of attack due to the aerodynamic deformation of the model, and the numbers described in the table are those at $\alpha = 2.94$ deg. The wing upper surface and the tail is covered by the finest level of the grid and the other part is covered with the second finest grid. The grid is generated on a work station (Xeon E5-2643 v3 at 3.4 GHz) using 1 core. The wall clock time and memory usage for the generation of the medium grid are 43 minutes and 50 GB, respectively.

In addition, reference solutions are computed using JAXA's unstructured flow solver FaSTAR²³ on the body-fitted structured grids that are provided in the APC-III workshop (see Fig. 8). The calculation by FaSTAR is conducted at $\alpha = -0.62, 2.94, 4.65, 5.72$ deg on the medium grid (9,006,808 cells), and $\alpha = 2.94$ deg on the fine grid (30,397,977 cells).

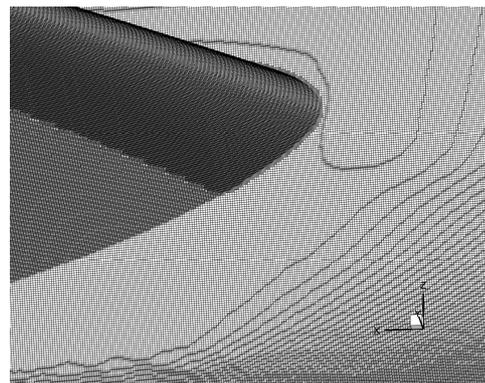
In both of the flow solver, the QCR-2000 correction²⁴ is used with the SA turbulence model. The reference²⁵ has reported that this correction improves prediction of the side-body separation at the high angles of attack. UTCart also uses the negative model¹⁸ to avoid instability of the calculation where $\tilde{\nu}$ become negative. The other calculation method is tabulated in Table. 2.



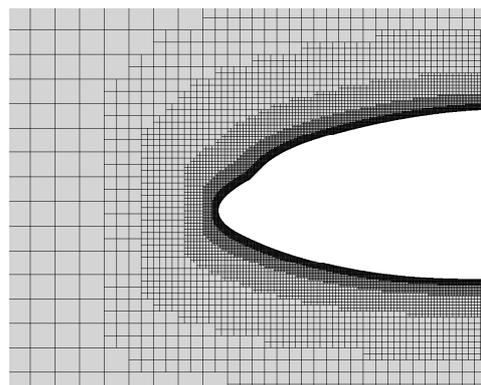
(a) Overview (very coarse grid for visualization, minimum grid size: 3.91 in)



(b) Main wing tip



(c) Main wing root

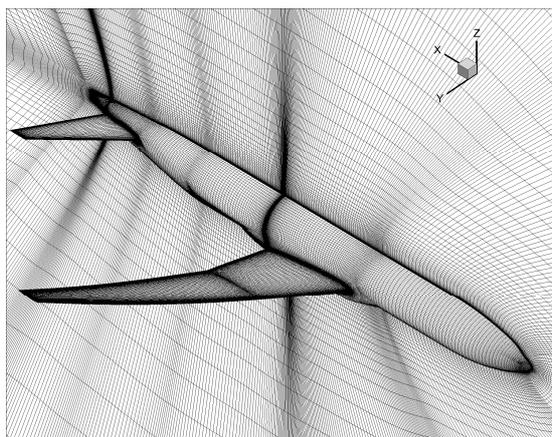


(d) Symmetry plane near nose

Fig. 7 Computational grid for UTCart (Medium grid except for the overview).

Table 1 UTCart grid settings around NASA-CRM.

	Coarse	Medium	Fine
Total cell number	24,415,860	50,323,727	97,041,807
Domain size in	4.80×10^4	3.60×10^4	5.40×10^4
Grid size (wing upper surface / tail) in	0.732	0.549	0.412
Grid size (wing lower surface / fuselage) in	0.366	0.274	0.206
Smooth layer (near field)	3	6	8
Smooth layer (far field)	3	3	3
Mean aerodynamic chord / Grid size (wing upper surface)	753	1,004	1,339

**Fig. 8 Overview of computational grids for FaSTAR (Medium Grid).****Table 2 Computational methods.**

Solver	UTCart	FaSTAR
Inviscid flux	SLAU	
Spatial Scheme (Inviscid term)	Second-order MUSCL	
Limiter	Barth-Jespersen ²⁶	Hishida ²⁷
Spatial Scheme (Viscous term)	Second order central difference	
Gradient Estimation	Weighted least-squares (G) ²⁸	GLSQ ²⁸
Time Integration	MFGS	LUSGS

V. Computational Results

The UTCart computation on the medium grid at $\alpha = 2.94$ deg requires 8000 iterations with a local time stepping at CFL = 100 to obtain the convergence. The magnitude of the oscillation of the drag coefficient in the final 1000 iterations is within 0.1 drag count (1 drag count = 10^{-4}). The computation was executed on the Reedbush-U supercomputer in the University of Tokyo (Xeon E5-2697 v4 at 2.1 GHz \times 2 per node) using 4 nodes (144 cores) with the MPI parallelization. The wall-clock time and memory usage for one medium grid case are approximately 5.5 hours and 60 GB, respectively.

A. Grid Convergence Study at $\alpha = 2.94$ deg

The grid convergence at $\alpha = 2.94$ deg is examined. At this condition, the lift coefficient is similar to the value at the cruise condition of the NASA-CRM ($C_L = 0.5$). Figure 9 shows the distribution of y^+ at the IP on the UTCart medium grid. Note that the surface values are extrapolated from the Cartesian grid onto the input surface CAD data. On the wing upper surface and the tail, the magnitude of y^+ is small compared with the other parts because the distance between the IP and wall is proportional to the grid size assigned to the parts. The maximum value of y^+ on the wing upper surface is 364. In the previous study on the immersed boundary method⁶, the distribution of the pressure on an airfoil can be accurately predicted on a grid that has a similar value of y^+ .

The pressure distributions on the surface in the results by UTCart on the medium grid and by FaSTAR on the medium grid are compared in Fig. 10. The qualitative feature of the distribution including the position and shape of

the shock on the wing upper surface shows fair agreement between the two results. The surface pressure distribution at the wing sections is visualized in Fig. 11. The sections are identical with the pressure tap position in the experiment, as visualized in Fig. 12. At section E, the thickness of the shock in the UTCart result is slightly smaller than that in the FaSTAR result. This is because the UTCart grid has higher chord-wise grid density than that of the grid for FaSTAR.

Figure 13 shows the component-wise aerodynamic coefficients. The pressure drag coefficient computed by the UTCart is overestimated compared with the FaSTAR result. The main causes of the overestimation are the wing and tail. Similarly, the pitching moment coefficient is underestimated in the UTCart result, although the trend of grid convergence is observed. These discrepancies may be due to the grid resolution of the leading edge/trailing edge. The grid convergence behavior of the total drag coefficient is shown in Fig. 14. The drag coefficient is plotted versus $N^{-2/3}$, where N is the total cell number. The FaSTAR result will converge to 307 drag counts. The linearly extrapolated value in the UTCart results using the medium and fine grids (the left 2 points) is slightly larger. However, the slope between the left two point in the UTCart result is steeper than from that between the right two point, and the grid converged value can be smaller than the extrapolated value in the figure. The differences between the values predicted UTCart and the grid converged value of the FaSTAR result are 35 (11%), 24 (8%) and 16 (5%) drag counts on the coarse/medium/fine grids, respectively.

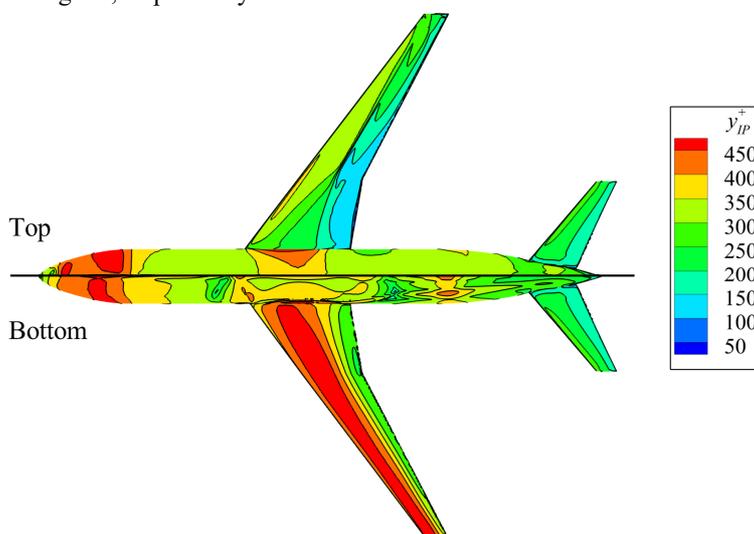
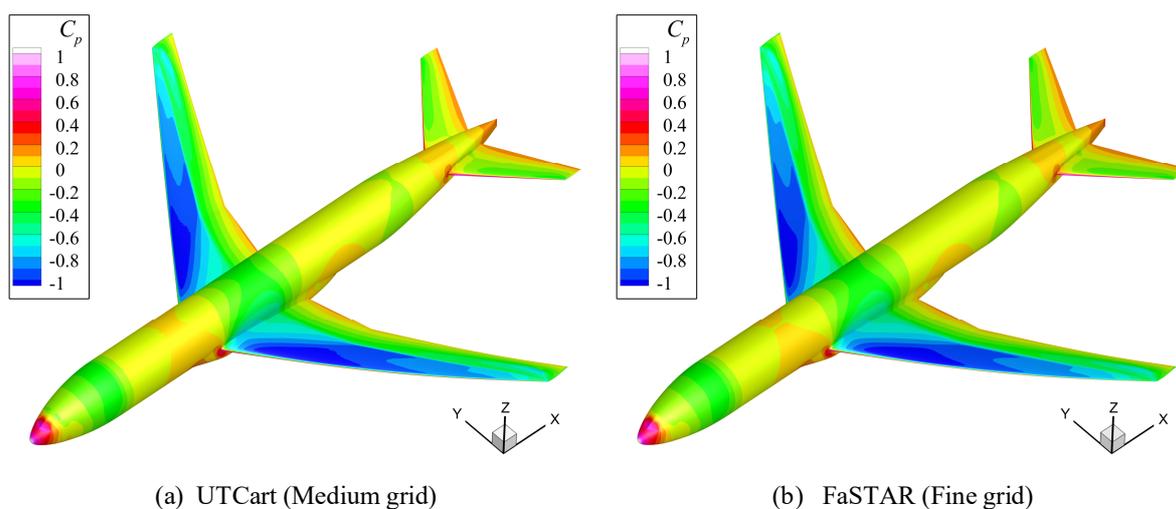


Fig. 9 Distribution of y^+ at the image point (Medium grid, $\alpha=2.94$ deg).



(a) UTCart (Medium grid) (b) FaSTAR (Fine grid)
Fig. 10 Distribution of surface pressure ($\alpha=2.94$ deg).

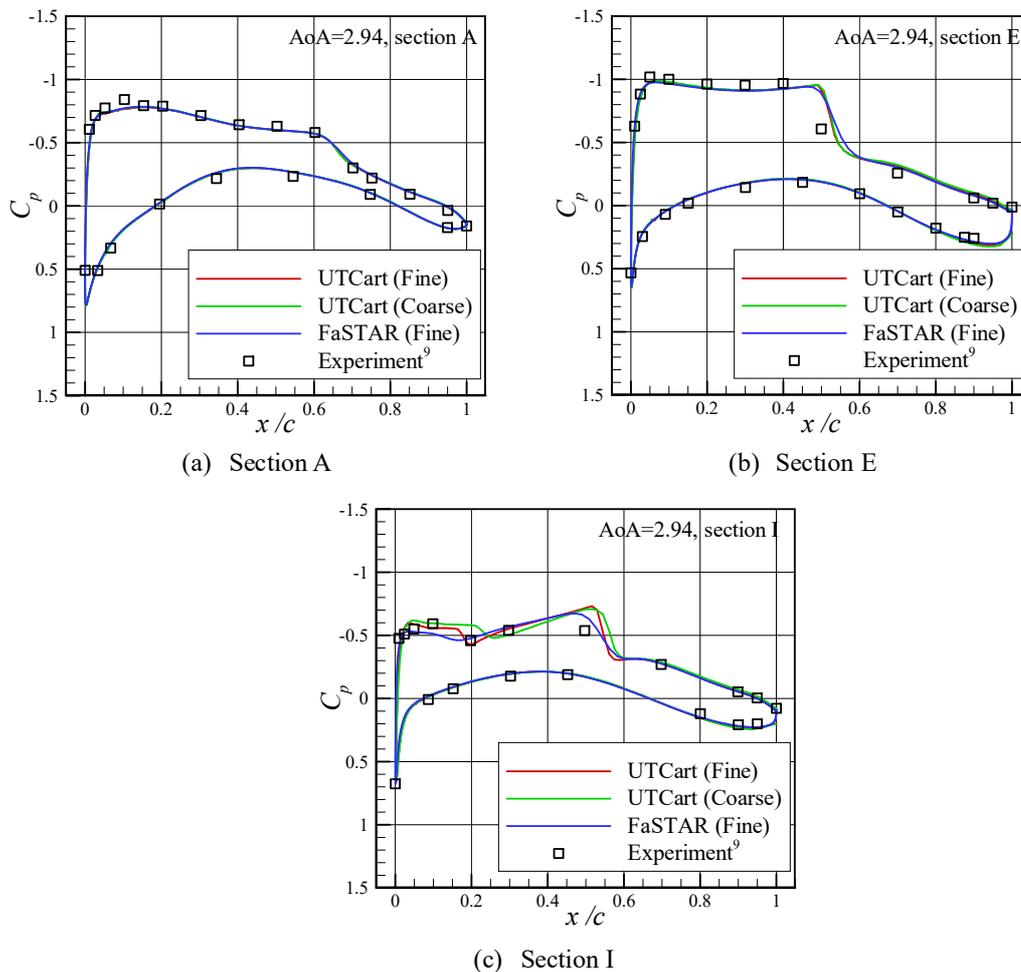


Fig. 11 Surface pressure coefficients at each section ($\alpha=2.94$ deg).

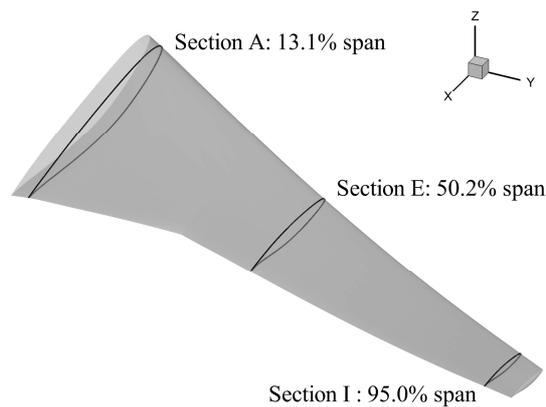


Fig. 12 Definition of the wing sections.

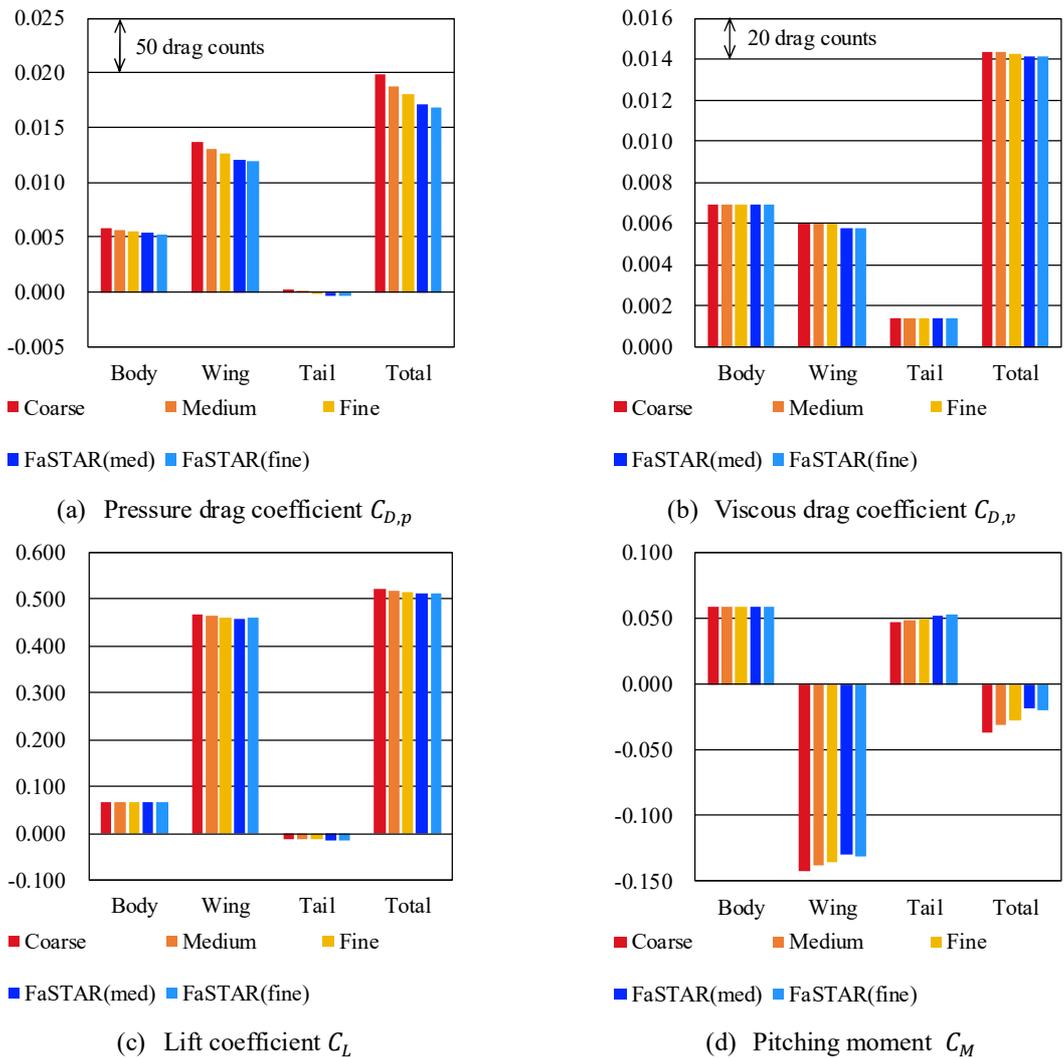


Fig. 13 Component-wise aerodynamic coefficients ($\alpha=2.94$ deg).

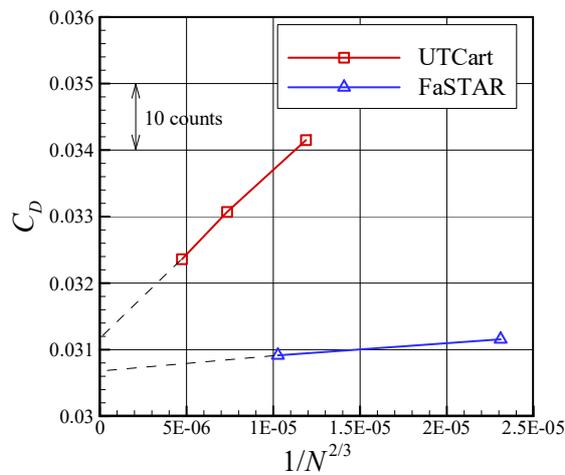


Fig. 14 Grid convergence of the drag coefficient ($\alpha=2.94$ deg).

B. α Sweep

Figure 15 shows the computed and measured aerodynamic coefficients at each angle of attack. The basic trend of each coefficient shows a fair agreement between the UTCart and FaSTAR results, and between the UTCart results and the measured data. In the computational results, however, the lift slope and the pitching moment coefficient is underestimated, while the drag coefficient is overestimated compared with the experimental data. The UTCart and FaSTAR results have these discrepancies in common, and thus they may be due to the difference of the configuration between the experiment and the computation (e.g. the support sting).

Figure 16 shows the surface pressure coefficient at $\alpha = 4.65$ deg. The difference of the pressure on the upper surface between the UTCart result and the experimental data is more visible than the $\alpha = 2.94$ deg case. It is because flow separation occurs near the wing-body junction and behind the shock as shown in Fig. 17. Although both flow solver reproduce a similar pattern of the flow separation, the area and the shape of the separated region is different. The separated region in the UTCart result is slightly smaller than the FaSTAR result. Also, the friction in the separated region is small in the UTCart result. This is because the wall function is used even in the separated region, and the boundary condition assumes the turbulent boundary layer attaches in the negative direction of the x -axis. The prediction of the flow separation is a continuing concern in the aerodynamic prediction of the NASA-CRM^{25,29-31} even on body-fitted grids, and the prediction accuracy needs to be investigated further with different grids or turbulence models.

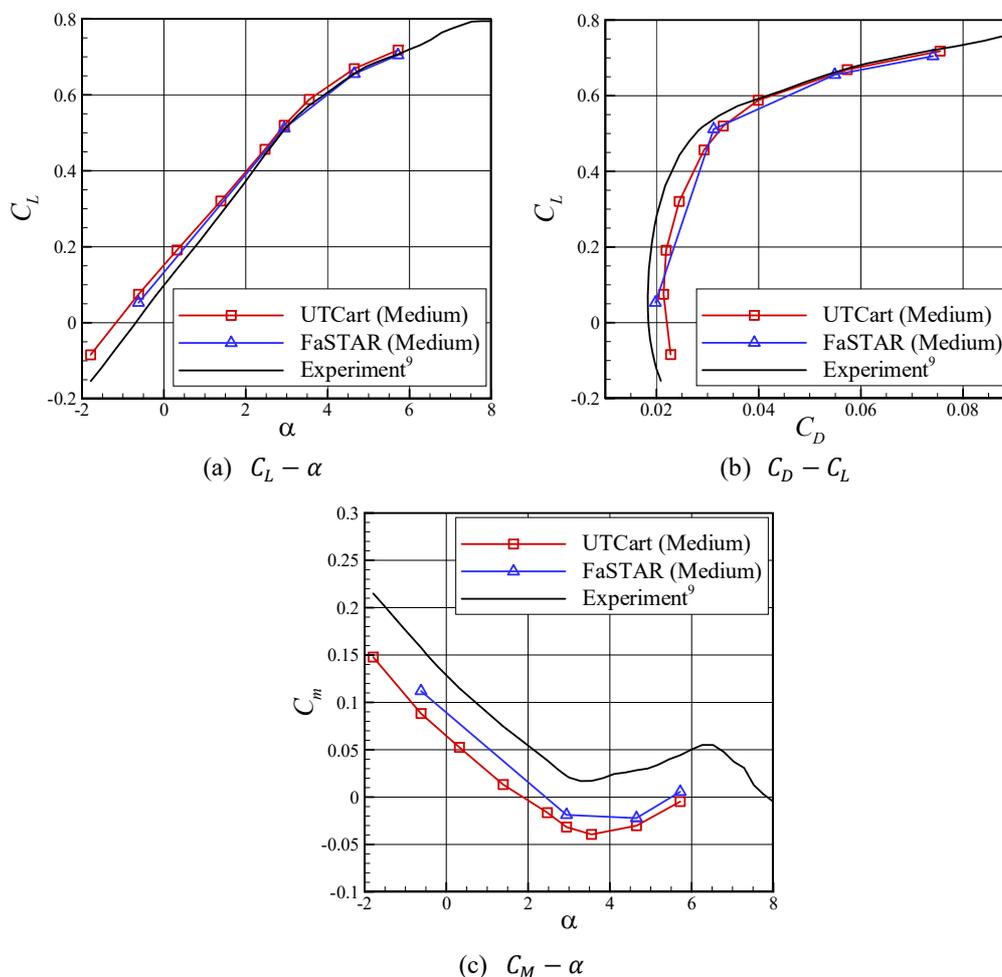


Fig. 15 Aerodynamic coefficients at each angle of attack.

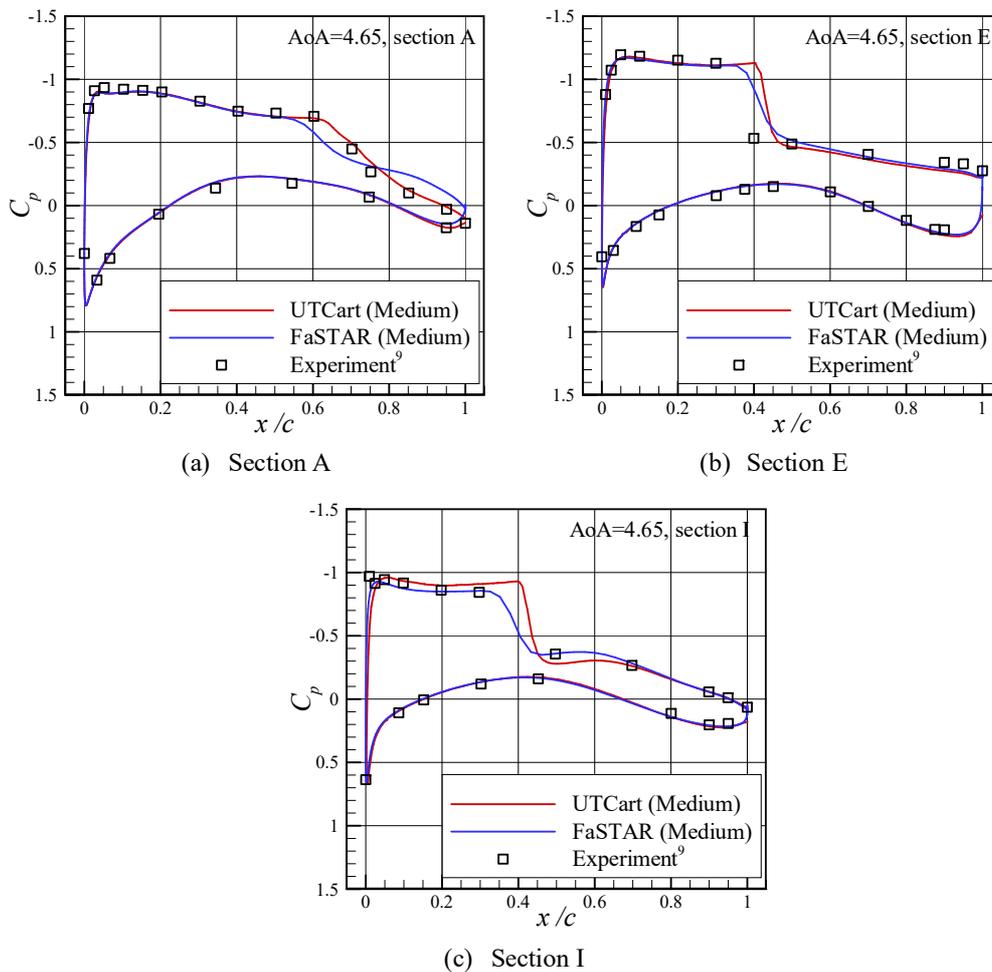


Fig. 16 Surface pressure coefficients at each section ($\alpha=4.65$ deg).

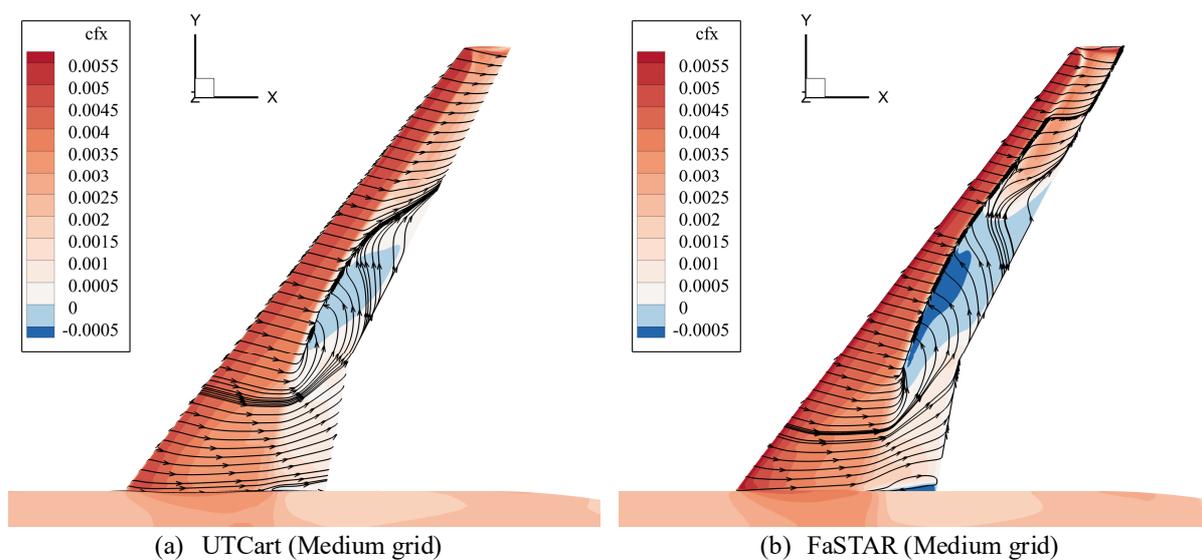


Fig. 17 Surface friction and streamline over the main wing ($\alpha=4.65$ deg).

VI. Summary

The transonic flows around the NASA Common Research Model was simulated to investigate the capability of the Cartesian-grid-based flow solver UTCart in 3D high Reynolds number flow simulations. Firstly, the framework of UTCart was explained. UTCart consists of an automatic Cartesian grid generator and a compressible flow solver. In the flow solver, the immersed boundary method with a turbulent wall function is used to reproduce the turbulent boundary layer. Some updates for the computational method was also explained. The relationship between the eddy-viscosity and the temperature near wall is clarified, and the boundary heat flux was updated. Furthermore, the calculation method of the cutoff distance that is required in the turbulent boundary condition was developed using an advection equation.

Using the UTCart grid generator, the medium grid (approx. 50 million cells) was generated in 43 minutes. On the medium grid, the y^+ on the wing upper surface is up to 364, and more than a thousand cells are located in the mean aerodynamic chord. The trend of each component-wise aerodynamic coefficient showed a consistent trend of grid convergence toward the values computed on a conventional body-fitted grid, although the drag coefficient was slightly overestimated. Compared with the grid converged value, the drag coefficient at the cruise condition had 24 (8%) and 16 drag count (5%) differences in the medium and the fine grid (approx. 97 million cells) results, respectively. Furthermore, the trend of the aerodynamic coefficients and the qualitative flow features including flow separation at high angles of attack showed a fair agreement with the experimental data and with the computational results on the body-fitted grids.

Appendix A. Surface Force Integration on Cartesian Grids

On the Cartesian grids, the cell boundary is different from the actual surface of the object. Thus, the aerodynamic force acting on the object is evaluated based on the balance of the numerical flux through the cell boundary. The method starts from the equation of the momentum conservation

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j + p \delta_{ij}) = \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \quad (\text{A.1})$$

where ρ is the density, u_i is the component of the velocity, p is the pressure, τ_{ij} is the component of the stress tensor, and f_i is the component of the body force. The aerodynamic force working on the object is calculated as a reaction of the force working on the fluid. Thus, integrating eq. (A.1) in the domain within the cell boundary Γ_G (see Fig. A), and applying the divergence theorem yields

$$F_i = - \int_{\Gamma_G} \{\rho u_j u_i + (p - p_\infty) \delta_{ij} - \tau_{ij}\} n_j dS - \int_{V_1} \frac{\partial(\rho u_i)}{\partial t} dV, \quad (\text{A.2})$$

where F_i is the aerodynamic force in i th direction, n_j is the component of the normal vector of Γ_G , and V_1 is the domain between Γ_G and the object. Here, the volume of V_1 is small, and the flow in this domain is assumed to be nearly steady. Thus, the unsteady term (the second term) in eq. (A.2) is neglected. Equation (A.2) also assumes that no source/sink of mass exists in V_1 . However, the immersed boundary method does not assure a strict mass conservation in V_1 . Therefore, a compensation term for the mass source/sink is added to (A.2) with assuming the mass recovers the free-stream velocity at the far-field:

$$F_i = - \int_{\Gamma_G} \{\rho u_j u_i + (p - p_\infty) \delta_{ij} - \tau_{ij}\} n_j dS + \dot{M} U_{\infty, i}, \quad (\text{A.3})$$

where \dot{M} is the mass source/sink (a source has a positive value), $U_{\infty, i}$ is the component of the free-stream velocity. The mass source/sink is evaluated by the integration of the continuity equation along Γ_G as

$$\dot{M} = \int_{\Gamma_G} \rho u_j n_j dS. \quad (\text{A.4})$$

Here, substituting (A.4) into (A.3) yields

$$F_i = - \int_{\Gamma_G} \{\rho(u_j - U_{\infty, j})u_i + (p - p_\infty) \delta_{ij} - \tau_{ij}\} n_j dS. \quad (\text{A.5})$$

In the flow solver, equation (A.5) is discretized on the faces that compose Γ_G as

$$F_i = - \sum_{\text{face}} [\{(\rho u_i u_j + p \delta_{ij}) n_j\} - \langle \rho u_j n_j \rangle U_{\infty, i} - p_\infty \delta_{ij} - \langle \tau_{ij} n_j \rangle] \Delta S, \quad (\text{A.6})$$

where $\langle (\rho u_i u_j + p \delta_{ij}) n_j \rangle$ and $\langle \rho u_j n_j \rangle$ are the components of the inviscid flux through the face, $\langle \tau_{ij} n_j \rangle$ is the component of the viscous flux, and ΔS is the area of the face. The fluxes in (A.6) must be the numerical fluxes used

in the flow calculation. Note that the integration of only the viscous flux represents the viscous component of the aerodynamic force, and the remainder is considered as the pressure component.

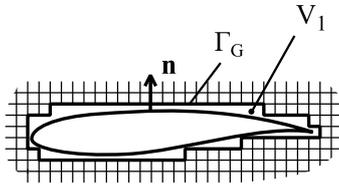


Fig. A Description of the domain and boundary topologies.

Acknowledgments

This research is funded by JSPS KAKENHI Grant Number 16J07740. The basic flow solver is the outcome of JSPS KAKENHI Grant Number 15H05559 (Grant-in-Aid for Young Scientists (A)). The authors are grateful to the Japan Aerospace Exploration Agency (JAXA) for permitting to use the fast unstructured flow-solver FaSTAR. The authors also thank the committee of the APC workshop for providing the computational grids for FaSTAR and the input geometry of the NASA-CRM.

References

- ¹Wang, Z. J. "A Quadtree-based adaptive Cartesian/Quad grid flow solver for Navier-Stokes equations," *Computers & Fluids* 27.4 (1998): 529-549.
- ²Hartmann, D. Meinke, M., Schröder, W., "A Strictly Conservative Cartesian Cut-Cell Method for Compressible Viscous Flows on Adaptive Grids," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 9, 2011, pp. 1038-1052.
- ³Lahur, P., "Automatic hexahedra grid generation method for component-based surface geometry," 17th AIAA Computational Fluid Dynamics Conference. 2005-5242.
- ⁴Berger, M. J., Aftosmis, M. J., and Allmaras. S. R. "Progress towards a Cartesian cut-cell method for viscous compressible flow," AIAA Paper, No. 2012-1301.
- ⁵Capizzano, F., "Turbulent Wall Model for Immersed Boundary Methods," *AIAA Journal*, Vol. 49, No.11, 2011, pp. 2367-2381.
- ⁶Tamaki, Y., Harada, M., and Imamura, T. "Near-Wall Modification of the Spalart-Allmaras Turbulence Model for the Immersed Boundary Method," *AIAA Journal*, 2017 (accepted).
- ⁷<http://turbmodels.larc.nasa.gov/> [retrieved on April 27th, 2017]
- ⁸Vassberg, J. C., DeHaan, M. A., Rivers, A. M., Richard, A. W., "Development of Common Research Model for Applied CFD Validation Studies," AIAA paper No. 2008-6919.
- ⁹Ueno M., Kohozai, M. and Koga, S., "Transonic Wind Tunnel Test of the NASA CRM: Volume 1," JAXA Research and Development Memorandum, JAXA-RM-13-017E, 2014.
- ¹⁰Spalart, P. R., and Allmaras S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA paper 92-0439.
- ¹¹Yoon, S. and Jameson, A., "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations," *AIAA Journal*, Vol.26, No.9, 1988, pp. 1025-1026.
- ¹²Shima, E., "A simple Implicit Scheme for Structured/Unstructured CFD," *Proceedings of 29th Fluid Dynamics Symposium*, 1997 (in Japanese).
- ¹³Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of computational physics* 43.2 (1981): pp. 357-372.
- ¹⁴Shima, E. and Kitamura, K., "Parameter-Free Simple Low-Dissipation AUSM-Family Scheme for All Speeds," *AIAA Journal*, Vol.49, No.8, 2011, pp. 1693-1709.
- ¹⁵Luo, H., Baum J. D., Löhner R., "A Hybrid Cartesian Grid and Gridless Method for Compressible Flows," *Journal of Computational Physics*, Vol. 214, No. 2, 2006, pp. 618-632.
- ¹⁶Tamaki, Yoshiharu, and Taro Imamura. "Efficient dimension-by-dimension higher order finite-volume methods for a Cartesian grid with cell-based refinement," *Computers & Fluids* 144 (2017): 74-85.
- ¹⁷Wang, G., Schöppe, A., and Heinrich, R., "Comparison and Evaluation of Cell-Centered and Cell-Vertex Discretization in the Unstructured TAU-Code for Turbulent Viscous Flows," *ECCOMAS CFD 2010*, Lisbon, Portugal, 2010.

¹⁸Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," 7th International Conference on Computational Fluid Dynamics, 2012.

¹⁹White, F. M. "Viscous Fluid Flow," McGraw-Hill, 2006.

²⁰Sussman, M., Smereka, P. and Osher, S., "A level set approach for computing solutions to incompressible two-phase flow," *Journal of Computational physics* 114.1 (1994): pp. 146-159.

²¹<https://aiaa-dpw.larc.nasa.gov/> [retrieved on April 27th, 2017]

²²<https://cfdws.chofu.jaxa.jp/apc/> [retrieved on April 27th, 2017]

²³Hashimoto A., et al. "Toward the Fastest Unstructured CFD Code 'FaSTAR,'" AIAA Paper, No. 2012-1075.

²⁴Spalart, P., R., "Strategies for turbulence modelling and simulations," *International Journal of Heat and Fluid Flow* Vol. 21, (2000): 252-263.

²⁵Yamamoto, K., Tanaka, K., and Murayama, M., "Effect of a Nonlinear Constitutive Relation for Turbulence Modeling on Predicting Flow Separation at Wing-Body Juncture of Transonic Commercial Aircraft," AIAA paper, No. 2012-2895.

²⁶Barth, T., and Jespersen, D., "The design and application of upwind schemes on unstructured meshes," AIAA paper, No. 89-0366.

²⁷Hishida, M. et al, "A new slope limiter for fast unstructured CFD solver FaSTAR," JAXA special publication: Proceedings of 42nd Fluid Dynamics Conference / Aerospace Numerical Simulation Symposium 2010, JAXA-SP-10-012, 2010 (in Japanese).

²⁸Shima, E., Kitamura, K. and Haga, K., "Green-Gauss/Weighted-Least-Squares Hybrid Gradient Reconstruction for Arbitrary Polyhedra Unstructured Grids," *AIAA Journal*, Vol.51, No.11, 2013, pp. 2740-2747.

²⁹Sclafani, A., DeHaan, M., Vassberg, J., Rumsey, C., and Pulliam, T., "Drag Prediction for the NASA CRM Wing-Body-Tail using CFL3D and OVERFLOW on an Overset Mesh," AIAA paper, No. 2010-4219.

³⁰Lee-Rausch, E. M., Hammond, D. P., Nielsen, E. J., Pirzadeh, S. Z., and Rumsey, C. L., "Application of the FUN3D Solver to the 4th AIAA Drag Prediction Workshop," *Journal of Aircraft*, 51.4, (2014), pp. 1149-1160.

³¹Sclafani, A., et al. "DPW-5 Analysis of the CRM in a Wing-Body Configuration using Structured and Unstructured Meshes," AIAA paper, No. 2013-0048.